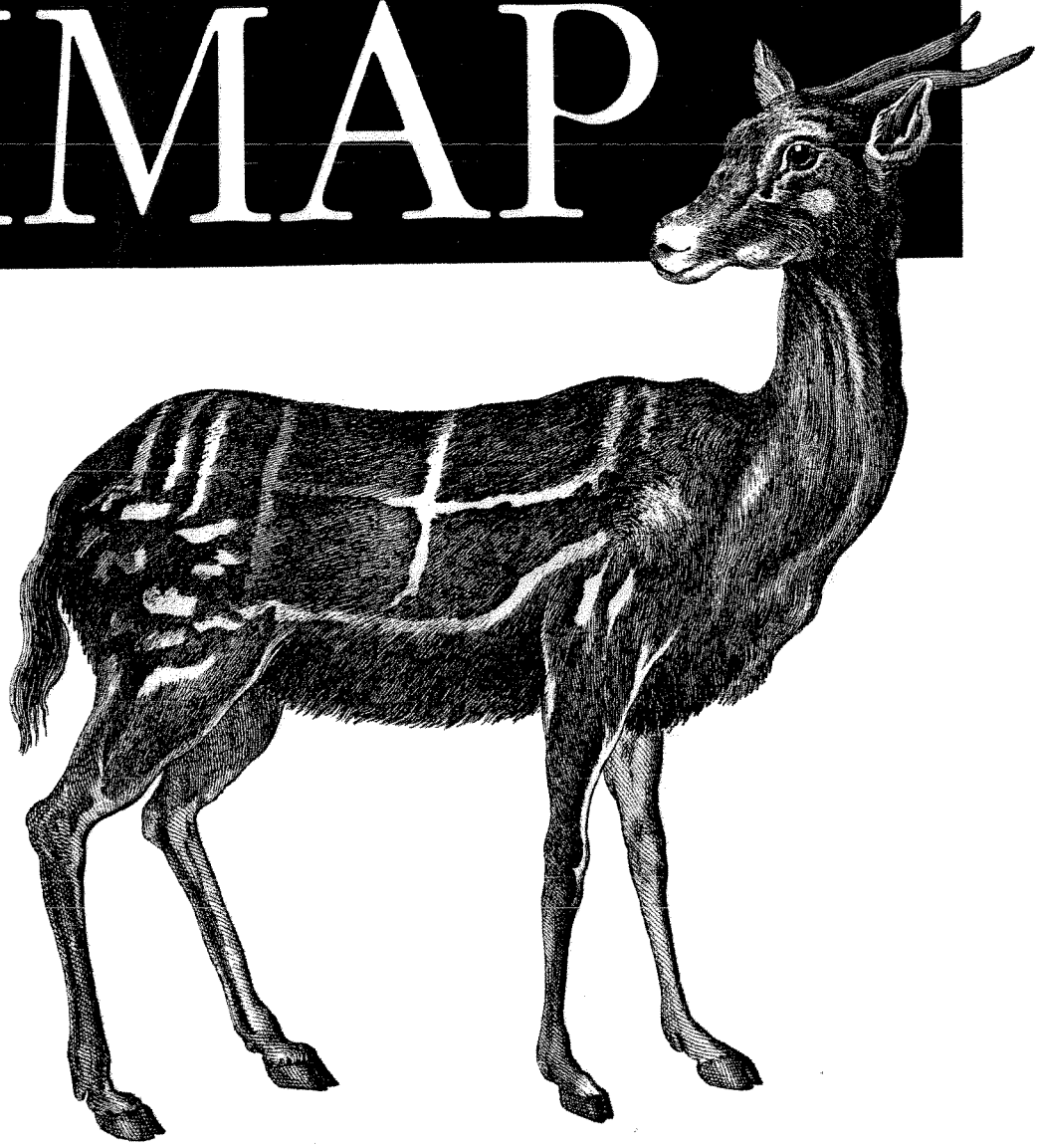
Help for Email Administrators

Managing IMAP



Dianna Mullet & Kevin Mullet
Foreword by Mark Crispin

O'REILLY®

Managing IMAP
by Dianna Mullet and Kevin Mullet

Published by O'Reilly & Associates, Inc., 101 Morris Street, Sebastopol, CA 95472.

Editor: Mike Loukides

Production Editor: Melanie Wang

Cover Designer: Ellie Volckhausen

Printing History:

September 2000: First Edition.

Library of Congress Cataloging-in-Publication Data

Mullet, Dianna
Managing IMAP/Dianna Mullet & Kevin Mullet.
p. cm.
Includes index.
ISBN 0-596-00012-X
1. Electronic mail systems 2. Internet programming. I. Mullet, Kevin.

TK5105.73 .M84 2000
005.7'1376--dc21 00-064970

ISBN: 0-596-00012-X
[M]

Foreword

For nearly 10 years after I invented IMAP, it was called "the best-kept secret in electronic messaging." Then, around 1995, key electronic messaging software vendors became convinced of its advantages over the older Post Office Protocol (POP), and IMAP's presence in the industry exploded.

Today, IMAP has evolved into a mature, widely deployed protocol. Even longtime proponents of POP have jumped on the IMAP bandwagon. More and more users are demanding IMAP, because IMAP is the only message access protocol that has the flexibility to accommodate their needs.

These users are part of a growing trend to access the Internet from a variety of different computers, appliances, and access paths. In one day, they may use a dialup from home; a wireless link on the bus, train, or ferry; and a super-high-capacity Internet2 link at the office. They may use a laptop while traveling and a desktop at home and work. They may be students using shared machines in a PC lab, with no personal data kept on the PC.

These users have more advanced needs for messaging than a single incoming mailbox that is periodically downloaded to a single computer. They may have a large archive of messages in a multitude of mailboxes that they need to reference at any time and in any place. They may participate in bulletin boards or other shared message collections (e.g., a shared "customer service" mailbox).

It could be any or all of the above. The common thread in these needs is highly diverse mail access patterns and needs. No matter how many different computers,

network connections, or mailboxes one uses, IMAP offers the flexibility needed for effective access.

The trend is clear. The word is spreading to those email users who are not yet familiar with IMAP's benefits. Corporate and ISP email providers are feeling the pressure to upgrade their services. Many server administrators, comfortable with a POP-only facility for many years, now find themselves playing catch-up.

Understanding the requirements of the task, much less planning how to do it, can be daunting. Fortunately, the "doing it" part is relatively painless once the way is pointed out.

Kevin and Dianna Mullet have done an outstanding job of presenting practical information on how to install, configure, and manage an IMAP server system. Both major freeware server implementations are covered in detail. I am particularly impressed by how the Mullets have uncovered and fleshed out useful information from mailing list folklore or passing mention in documenting.

A book like this has been needed for a long time. It belongs on every server administrator's bookshelf.

—Mark Crispin
Seattle, Washington
1 July 2000

It's not unusual for an IMAP mailbox to be hit by multiple IMAP clients running on different client machines simultaneously. For example, a user might have a program (like *biff*) that periodically polls her mailbox for new mail every five minutes. At the same time, she might leave herself logged in to the IMAP server via PINE 24 hours a day and check her mail from home using Outlook Express every evening. Not to mention the odd stop at the Internet café where she uses Mulberry to check her mail on her coffee break. Such a barrage of IMAP operations against an IMAP mail server's I/O resources should be considered commonplace.

Why Not Both POP and IMAP?

All the discussion of the relative merits of POP and IMAP and how they are appropriate for different sets of users begs the question: why not do both? This is, in fact, a quite reasonable proposition.

If you run a UW or Cyrus IMAP server, you're in luck—both servers come with a POP server that natively accesses the same mailstore as the IMAP server. Another way to do both is to run a POP-to-IMAP proxy (one comes with the UW server). The POP-to-IMAP proxy never talks directly to your mailstore. It simply translates the POP protocol into an IMAP stream, which is then directed at your IMAP server. In fact, if you have a dug-in POP-ulation that you'd like to convert to IMAP, you could just replace your production POP server with a POP-to-IMAP proxy and tell your users, "By the way, we also offer IMAP if you'd like to try it on for size." That is not an altogether uncommon strategy. The downside of using a POP-to-IMAP proxy is that you end up supporting both POP and IMAP clients.

Fortunately, POP and IMAP clients are increasingly becoming the same software—it's just a matter of configuring them differently.

Advantages of IMAP

Here's the rundown on advantages that are unique to IMAP.

Appending to mailboxes

We implied it earlier in this chapter, but it bears repeating. IMAP may be used not only to retrieve messages from your remote mailbox, but also to add them to your remote mailbox. Again, IMAP operates on email much like a database, with the messages being individual records. Using an IMAP client, a user can freely move messages about between his INBOX and additional remote mailboxes of his own creation.

Many of the IMAP users in the future will probably have no idea that IMAP is what makes their extended messaging environment possible. Here's where IMAP is headed:

- As universal multi-device connectivity becomes more common, IMAP will surely play a central role in coordinating simultaneous access to users' mailboxes from many devices at one time. POP simply doesn't scale in that regard.
- An unexpected wrinkle in the story of IMAP has been its growth as a back-office protocol supporting web-based email frontends (see Chapter 5, *Web-Based IMAP Clients*). IMAP's importance will probably be equally split between back office support of web-based frontends and direct interaction with client user interfaces.
- If the activity surrounding the development of extensions to IMAP is any indication, we can look forward to IMAP having a productive life, far exceeding that of POP.
- Finally, as the Internet moves into an age when ever fewer users need to know that such things as IP addresses, netmasks, and mail access protocols exist, IMAP's destiny may be that of the "man behind the curtain."*

Open Source Server Implementations

For nearly a decade, there have been only two appreciable open source IMAP servers: the University of Washington's IMAP server and Carnegie Mellon University's Cyrus IMAP server. Both are robust, time- and user-tested servers with an install base to rival any commercial alternative.

We will go into more detail in later chapters about the interesting histories and specific features of each server, but let's take a brief look at them for the time being.

University of Washington IMAP Server

The University of Washington server is the reference implementation of IMAP. It was written by Mark Crispin, the inventor of IMAP. It was started in 1988 at Stanford University as a C rewrite of the original Interlisp client and DEC-20 assembly language server. When Crispin changed jobs for the University of Washington late in 1988, the IMAP project went with him.

The University of Washington IMAP server strives for compatibility with existing Unix systems. If you've stored your mail in a given format on a Unix system over

* A reference to *The Wizard of Oz*.

the course of the past few decades, chances are that the UW server can read that format out of the box.

The UW server has a number of interesting experimental extensions, such as multiple append and server-based sorting and threading. The UW server also supports an extensive list of international character sets.

The UW server is very modular—it is easy to add support for another mailbox format or SASL authenticator by writing a code module and relinking.

Carnegie Mellon University Cyrus IMAP Server

The CMU Cyrus Server is a component of a project called Project Cyrus.* Started in 1994, the Cyrus project was started because the management overhead of running the existing proprietary system was getting to be too high. CMU was unable to keep up with client development in particular and wanted to use commercial off-the-shelf (COTS) mail clients. Project Cyrus was created to provide a next-generation messaging system that relied heavily on Internet standards and was highly scalable as well as modular, supported disconnected mode, and enabled freedom from legacy architecture. Cyrus was a rejection of the idea of basing an email system on software that just copied files around from place to place in a file-system. Historically, the lack of reliability of such systems has only been exceeded by their lack of scalability.†

Although the core of the Cyrus project is the Cyrus IMAP server, CMU has developed the related protocols IMSP and ACAP, and implementations of those protocols to support their IMAP server, as well. The IMSP and ACAP servers offer a way to store the user's personal client settings remotely. The SASL library provides a way for any Internet standard-based application (client or server) to perform Internet standard authentication. CMU's Cyrus IMAP server also implements the SIEVE server-side filtering language, to boot.

Perhaps its greatest quality is the Cyrus server's feature richness. Given the right client, users benefit from such features as support for IMAP quotas and mailbox access control lists. The University of Washington server is rich with support for differing types of email storage formats. If you've stored your email in a given format on a Unix system over the course of the past few decades, chances are that the UW server can read that format.

The difference between the UW and Cyrus servers can be summarized as follows: UW is a generalist and Cyrus is a specialist.

---

* *http://asg.web.cmu.edu/cyrus/.*

† Not that people learn, however. Many popular LAN-based proprietary email systems are, under the hood, little more than file-copying utilities. Internet email standards rule.

The UW server is engineered to be completely compatible with Unix mail. It assumes the worst case: a site where the mail server also acts as an interactive login server for the general population and where users also access their mail directly with host-based mail clients. The UW server does not add any management overhead—all quota and access control is handled by the kernel (e.g., with Unix file and directory permissions).

Cyrus is engineered to run on so-called "black box" servers, where the user's mail is read by no software other than the Cyrus server. Cyrus offers rich support for IMAP quotas and mailbox access control lists and a great deal of IMAP-specific management control. Cyrus also obtains considerable performance benefit from not having to be compatible with ancient email software.

Midrange-to-large IMAP sites that choose UW generally modify it in various ways to make it more of a specialist, like Cyrus. A number of large IMAP sites do use UW; usually these are large sites that don't want the management overhead of Cyrus.

IMAP-Related Standards and Documents

All the standards and related documents in this section are Request For Comment (RFCs) documents. There is a plethora of IMAP-related Internet Drafts (would-be standards that are still working their way through the approval process), but they change so rapidly that they won't be mentioned here. Everyone has favorite RFC archives. Here are ours. Each of the sites lets you choose the geographically closest archive from which to retrieve your documents:

- Internet RFC documents (*http://www.nexor.com/info/rfc/index/rfc.htm*)
- Internet Drafts (*http://www.nexor.com/info/internet-drafts/id.html*)

A word about these documents—they're the epitome of "hit the ground running" docs. They're meant to be terse and narrow, much like their Unix manpage cousins. Despite their laissez-faire name, RFCs are the canonical standards documents of the Internet. If you have a bet with someone and need an indisputable source to settle the argument, turn to the RFC.

Table 2-2 is a snapshot of the current RFC standards related to IMAP. For a more comprehensive list, you can do a database search on "IMAP" at one of the previously mentioned URLs. In Table 2-2, the most important document is the *Internet Message Access Protocol Version 4rev1*, RFC 2060, by Mark Crispin. Consider it the defining document of the core features of IMAP. Second behind it would be RFC 2683 (*IMAP4 Implementation Recommendations*), which is necessary to read to understand the IMAP "folklore."

In this chapter:

- *History of Cyrus*
- *Cyrus Concepts and Features*
- *Cyrus Server Configuration*
- *The Future of Cyrus*
- *Strengths and Weaknesses of Cyrus*
- *When Is Cyrus the Right Choice?*

6

Introduction to the Cyrus IMAP Server

This chapter provides a technical overview of the features and design concepts that make up the Cyrus IMAP server. The Cyrus server wasn't the first IMAP server, nor was it written by the author of the IMAP standard. True believers in the Cyrus server and its derivatives are likely to tell you, though, that once you've deployed the Cyrus server, you're unlikely to switch to another server. Cyrus boasts a very attractive feature set. It offers robust administration, scalability, and leading-edge IMAP extensions, such as mailbox quotas, plug-in authentication mechanisms, and support for server-side filtering. Detractors are likely to say that UW can be deployed more easily and that Cyrus may not make much sense for a small or medium-size user base.

The Cyrus IMAP server is based on IETF standard protocols, including IMAP4, IMSP, SMTP, RFC 822, MIME, and SASL. IMAP2bis is also supported for backward compatibility with earlier IMAP clients. POP3 is included to support POP users while they're going through the process of selecting an IMAP client.

The Cyrus server is feature-rich, and implements several IMAP protocol extensions, including the IMAP QUOTA and the IMAP ACL extensions. Cyrus is normally run as a "black box" server—users are not meant to access the system by any means other than the IMAP protocol. Mailboxes are stored in a central location and in parts of the filesystem that are private to the Cyrus server. Cyrus was designed with centralized mail storage in mind to make the system easier for administrators to manage. The centralized mailstore also allows such features as mailstore partitioning (discussed later in this chapter). The ability to partition the mailstore makes the server very scalable—as the mailstore grows, the system administrator may add new partitions to the mailstore to accommodate new growth, without affecting the operation of the server. Cyrus shared folders are very

flexible—the server not only allows concurrent read connections to the same mailbox, but also supports concurrent write connections.

The Cyrus server differs from other IMAP server implementations in that it is meant to run on sealed, or black box, servers, where normal users are not permitted to log in. Users access their mail strictly via IMAP. Even if a user were allowed to log on to the server where Cyrus runs, she would not be able to do things like "grep" her mail folders for someone's username or compress her folders. That is because the Cyrus mailstore is private to the Cyrus IMAP system. The private mailbox design can be difficult for users to get used to, especially if in the past they've logged in to a Unix shell account and had direct access to mail. Cyrus's black box design, however, gives the server large advantages in efficiency, scalability, and ease of administration.

Cyrus did not turn out to be fantastically scalable by accident. An original goal of Project Cyrus was to produce a mail system that scaled to tens of thousands of active mail readers.* The number of large† sites using Cyrus (Carnegie Mellon University, the University of North Texas, North Carolina State University, the University of Florida, Ohio University, the University of Otago in New Zealand, for example) are examples of Cyrus's scalability. There are even three IMAP server products released commercially—by Netscape, Mirapoint, and MessagingDirect—which are built on the original Cyrus distribution. Although Cyrus is suitable for large organizations, it can be used equally well in small organizations. Although not as simple to deploy as the UW server, it is capable of being installed, configured, and maintained by small computing departments.

History of Cyrus

The Cyrus IMAP server traces its roots back to CMU's Andrew Messaging System (AMS). AMS, the world's first large-scale multimedia mail and bulletin board system, was a very successful proprietary system developed and used at CMU to support the electronic communication needs of the university community. Project Cyrus began in 1992 in an effort to develop a replacement for AMS, which by then was reaching its end-of-life. The first Cyrus IMAP4 server was released in late 1995, one year after IMAP4 was approved as a Proposed Internet Standard. The Cyrus server retained many of the same features that made AMS a success.

---

* Project Cyrus was named after Cyrus the Great of Persia (599–530 BC), who initiated one of the first known postal systems.

† A large site is a site with more than 20,000 active users.

format "v2000.NNN" (UW IMAP 2000), it's most likely the UW server. The SCAN, SORT, and THREAD capabilities are UW experimental capabilities; their presence also suggests that the server is the UW server.

You can also look for artifacts of the UW *phile* driver. The *phile* driver is likely to be found only in software based on the UW C-Client library:

```
% strings /usr/local/sbin/imapd | egrep phile
phile
phile
phile recycle stream
%
```

Be careful, though, because there are likely to be other IMAP servers out there based on the C-Client libraries.

Probably the most interesting and significant fact about the UW IMAP server is that it was written by Mark Crispin, the progenitor of IMAP itself. It's fair to say that Crispin is to the IMAP community as Linus Torvalds is to the Linux community. Crispin invented IMAP entirely on his own, when he was asked to build a distributed mail system with no guidance. He wrote the original IMAP server from scratch in DEC-20 assembly language in 1985. IMAP's early design was strongly influenced by the DEC-20 mail system, of which Crispin was also the primary developer and maintainer. The first nine years of IMAP's development can be attributed entirely to Crispin.

History

The UW IMAP server was first written in November 1990. It took Mark Crispin only a few days to write *imapd*, because he based it on the C-Client library. Quoting Crispin, "If you have the right underlying structures and tools, any project can be reduced to triviality."

imapd didn't include support for traditional Unix mailbox format in its original release. The users of legacy Unix mail programs such as */bin/mail* were clamoring for Unix mailbox support in *imapd*, though. Crispin was reluctant to add Unix mailbox support because of its limitations, such as the inability to have a mailbox open by multiple users simultaneously. He added this support to a later release, though, and support for other mailbox formats followed soon after. The current preferred mailbox format, mbx, continues this tradition.

It's interesting that, from its inception, UW *imapd* supported multiple, simultaneous access to a single mailbox using tenex format, something that was considered impossible on Unix systems.